PHASE BASED MODELING OF DYNAMIC TEXTURES AND ITS APPLICATIONS

BY

BERNARD SEMAAN GHANEM

B.Eng., American University of Beirut, 2005

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2008

Urbana, Illinois

Adviser:

Professor Narendra Ahuja

ABSTRACT

Temporal or dynamic textures (DT's) are video sequences that contain a spatial texture undergoing stationary stochastic changes with time. These changes add a new dimension to the already present spatial stochastic repetitiveness. Thus, DT's are spatiotemporal extrapolation of image texture. Examples of DT's include moving water, foliage, smoke, clouds, etc. This paper presents a model of spatiotemporal variation characteristic of a DT. Most recent work on DT modeling represents the frames in a DT as the responses of a linear dynamical system (LDS) to noise. Despite its merits, this model has limitations because it attempts to model temporal variations in individual pixel intensities which do not take advantage of global motion coherence. We propose a new model that relates texture dynamics to the variation of the Fourier phase, which captures the relationships among the motions of all pixels (i.e. global motion) within the texture, as well as the fixed appearance of the spatial texture within each frame. Unlike LDS, our model does not require segmentation or cropping to eliminate any non-DT parts of the image frames for training, which allows it to handle DT sequences containing a static background. We test the performance of this model on three applications: DT synthesis, DT recognition, and DT video compression. Experiments with a dataset (available at [1]) that we have compiled demonstrate that our phase based model outperforms LDS in synthesis and recognition, while its compression performance exceeds that of MPEG-2.

This thesis is dedicated to my loving parents, brother, and sister. May it always see them in the best of health and prosperity.

ACKNOWLEDGMENTS

Working long hours on this thesis has helped me truly understand what Gibran Khalil Gibran meant in the following quote excerpted from his masterpiece, "*The Prophet*".

> If you cannot work with love but only with distaste, it is better that you should leave your work and sit at the gate of the temple and take alms of those who work with joy. And when you work with love you bind yourself to yourself, and to one another, and to God.

I would like to thank all people who have helped and inspired me during my graduate study. They are many, so I apologize beforehand if I absentmindedly forget to mention some. I thank The Lord for all His blessings, which I experience every day of my life. He has granted this undeserving servant so much.

My deepest gratitude goes to my beloved family for their unwearied love and support throughout my life. This thesis would not have been possible without them. I am eternally indebted to my father, Semaan, and my mother, Rana, who have never hesitated in sacrificing what they had to provide for their children. The love and respect they show me is only exceeded by the love and gratitude I have for them. May they lead long and healthy lives, through which I can always pride myself in being their son. I would like to thank my brother, Elie, and my sister, Simona, whose continuous support and encouragement were essential in this endeavor. God willing, I intend to continue reciprocating the same love and wise advice to them. I thank the rest of my family for their love, support, and encouragement. Special thanks to all my friends for their friendship and co-laboring.

Last but not least, this thesis would not have been possible without constant and invaluable help from my academic and research adviser, Professor Narendra Ahuja. The most important things Professor Ahuja has taught me are not written in these pages. I will cherish them always, till the end of my days. I no longer see him as just my adviser, mentor, and teacher, whose advice and counsel I consider dear. I am honored to have him as a friend.

TABLE OF CONTENTS

LIST OF FIGURES . . . . . **vii**

CHAPTER 1 INTRODUCTION **1**

CHAPTER 2 MOTIVATION **5**

CHAPTER 3 PROPOSED MODEL **9**
3.1 PDPP Algorithm 11
3.2 Non-Parametric Probabilistic PDPP Model 13
3.2.1 MAP-Based DT Synthesis 13
3.2.2 MAP-Based DT Recognition 14
3.3 Piecewise Smooth PDPP Model 15

CHAPTER 4 DT VIDEO COMPRESSION **17**
4.1 PDPP Compression 17
4.2 LDS Compression 18

CHAPTER 5 EXPERIMENTAL RESULTS **20**
5.1 APPLICATION (I): DT Synthesis 20
5.2 APPLICATION (II): DT Recognition 22
5.3 APPLICATION (III): DT Compression 27

CHAPTER 6 CONCLUSIONS **29**

CHAPTER 7 FUTURE WORK **30**

REFERENCES **31**

LIST OF FIGURES

2.1 (a) shows a schematic of the iterative phase-only reconstruction algorithm described in [2]. This algorithm consists of a sequence of FFT and IFFT operations, where the phase response of the initial image ($t = 0$) is kept constant. In (b), this algorithm is applied to a hybrid image ($t = 0$), whose phase response comes from an ocean image and the amplitude response from a fire image. We show the intermediate results of this algorithm for iterations $t = 50, 100, 250$. Note that the reconstructed image significantly resembles the ocean image, from which the phase response was extracted. . 7

2.2 Percentage of PCA components required to capture 80% of the variation in DT phase and DT amplitude, for a set of DT video sequences. For each DT, it is clear that significantly more PCA components of the DT phase are needed than DT amplitude. This substantial discrepancy is due to the fact temporal variations of DT phase are much more significant than those of DT amplitude. 8

3.1 Schematic plot of BPP (A, B, and C) and PDPP (A, $B^{'}$, and $C^{'}$) features in 2D. Transforming BPP features into PDPP features expands the embedding range space by 2π in each dimension. Consequently, the number of PCA components needed to represent the PDPP space is, in general, less than the number needed to represent the BPP space. In this example, the BPP range space requires two PCA components, while the PDPP space requires only one. 10

3.2 (a) and (b) plot the PDPP and BPP cumulative variances represented by different numbers of PCA components for fire and clouds sequences. It is clear that the PDPP model represents much more cumulative variance than BPP for the same number of PCA components. 11

3.3 Temporal variation of a PDPP feature component modeled by a cubic spline. 16

5.1 Sample synthetic images produced by the Cubic Spline (top three rows) and MAP (bottom three rows) synthesis methods. Videos of these sequences can be viewed at [3]. 21

5.2 Images (a), (b), and (c) are respectively a frame from the original sequence, a MAP synthetic frame, and a synthetic LDS frame. 23

5.3 (a) and (b) are portions of the confusion matrices for recognition of column DT as row DT by PDPP and LDS respectively. Blank entries correspond to no decisions. 26
5.4 (a)-(c) compare the mean compression performance of LDS, BPP, and PDPP for fire, ocean, and smoke DT video sequences. In (d), we show the temporal variations of LDS, BPP, and PDPP compression schemes for the fire sequence. In (e)-(h), we compare the compression performance of our PDPP algorithm to that of MPEG-2 encoding for a compression rate of approximately 60%. 28

CHAPTER 1

INTRODUCTION

A dynamic texture sequence (DT) captures a random spatiotemporal phenomenon. The randomness reflects in the spatial and temporal changes in the image signal. This may be caused by a variety of physical processes, e.g., involving objects that are small (smoke particles) or large (snowflakes), or (semi) rigid (grass, flag) or nonrigid (cloud, fire), moving in 2D or 3D, etc. Even though the overall global motion of a DT may be perceived by humans as being simple and coherent, the underlying local motion is governed by a complex stochastic model. For example, a scene of "translating" clouds conveys visually identifiable global dynamics; however, the implosion and explosion of the cloud segments during the motion result in very complicated local dynamics. Irrespective of the nature of the physical phenomena, the objective of DT modeling in computer vision and graphics is to capture the nondeterministic, spatial and temporal variation in images. DT modeling is motivated by a range of applications including DT synthesis, background subtraction in dynamic environments, and multi-layer motion separation.

The challenges of DT modeling arise from the need to capture the intrinsic visual properties of the large number of objects involved, their complex motions, and their intricate interactions. A good model must accurately and efficiently capture both the appearance and global dynamics of DT.

Related Work

The majority of methods that model DT fall into three broad categories which we briefly review next. **(1)** Motion field methods (e.g. [4],[5]) are based on motion analysis algorithms, such as those that compute and model optical flow. They are convenient, since frame-to-frame estimation of the motion field has been extensively studied and computationally efficient algorithms have been developed. However, these methods are best suited to estimate local and smooth motion fields. The non-smoothness, discontinuities, and noise inherent to rapidly varying, non-stationary DT's (e.g. fire) pose a challenge to optical flow algorithms. Object tracking methods (e.g. [6]) also tend to be infeasible here due to the large number of extremely small and non-rigid moving objects with little shape stability, complex motion characteristics, and inter-object interactions.

(2) Physical modeling methods (e.g. [7]) attempt to capture the attributes of the physical process from first principles. These methods are primarily used to synthesize specific textures such as ocean water, smoke, etc. Being closely tied to specific physical processes, they are difficult to generalize to other DT's. They are also computationally expensive since they must model physical phenomena.

(3) The third category consists of methods that obtain statistical models of spatiotemporal interdependence among images. They include the time series model of Mccormick et al. [8], the spatiotemporal auto-regressive (STAR) model by Szummer et al. [9], and multi-resolution analysis (MRA) trees by Bar-Joseph et al. [10].

The previous methods suffer from the following shortcomings: **(**i**)** DT representation limits the amount of data involved (e.g. only a finite-length sequence can be synthesized from the original DT). **(**ii**)** Constraints are imposed on the types of motion that can be modeled (e.g. neighborhood causality is imposed in both the spatial and temporal domains). **(**iii**)** Constraints are applied directly to pixel intensities instead of more succinct representations, thus, making them computationally more challenging and sometimes in-

feasible. Within this class of DT models, we mention the recent work of Doretto et al. [11] that derives a stable linear dynamical system (LDS) model for DT's. Consecutive frames of a DT sequence are linearly related and viewed as the responses of the LDS to random noise input. This model has been applied to DT synthesis [11], recognition [12], and segmentation [13]. In [14], LDS has been expanded to accommodate a mixture of modeled DT's and its computational complexity has been improved in [15]. Modifications that have been made to this method include incorporating a lower dimensional representation by using high energy Fourier descriptors or state space variables instead of the estimated model parameters [15]. In [16], multiple local LDS models were embedded into a graphical model framework, in which DT modeling was performed. Also, the LDS model has been applied to DT sequence registration in [17].

However, its modeling of the intensity values of a DT as a stable, linear ARMA (1) process leads to three main disadvantages: (*i*) the assumption of second-order probabilistic stationarity, which does not hold for numerous sequences (e.g. fire), (*ii*) the suboptimal relationship between the order of the LDS model and the extent of temporal modeling possible (i.e. an LDS of order n does not capture the most temporal variation in a DT among all models of order n), and (*iii*) significant computational expense, since the model is applied directly to pixel intensities without appropriately mitigating spatial redundancy.

Our method is based on a spatiotemporal, image-based model that uses the Fourier phase content of the DT sequence to capture both its appearance and global dynamics. In what follows, we justify our choice of using Fourier phase (Chapter 2), present the details of our phase based model (Chapter 3), and describe its use for DT synthesis and DT recognition. In Chapter 4, we describe how this model can be applied to DT video compression. Furthermore, Chapter 5 provides experimental results that compare the performance of our model to that of LDS with respect to DT synthesis, recognition, and compression. In this chapter, we also compare our compression scheme to that of MPEG-2 encoding. Finally,

we provide some final conclusions about our proposed model and highlight future research goals.

CHAPTER 2

MOTIVATION

In this chapter, we will establish that a model which better captures the appearance and dynamics of a DT than previous methods can be defined by representing its Fourier phase content alone. Following are the advantages of using the frequency domain representation that alleviate certain problems encountered in the spatial domain and motivate our proposed approach. **(1)** Spatially global features are captured locally in the frequency domain, since the change of the amplitude or phase of a certain frequency results in a global spatial variation. This makes frequency space modeling more appropriate for modeling global patterns such as those associated with DT appearance and dynamics. **(2)** Frequency analysis has been shown to be robust to unavoidable effects in images such as illumination changes [18] and additive noise [19]. **(3)** Computational complexity can be reduced by exploiting the inherent conjugate symmetry of the Fourier transform and the usually observed concentration of spectral image energy at low frequencies. **(4)** Efficient algorithms and specialized hardware are available for fast computation of the Fourier transform (e.g. FFT).

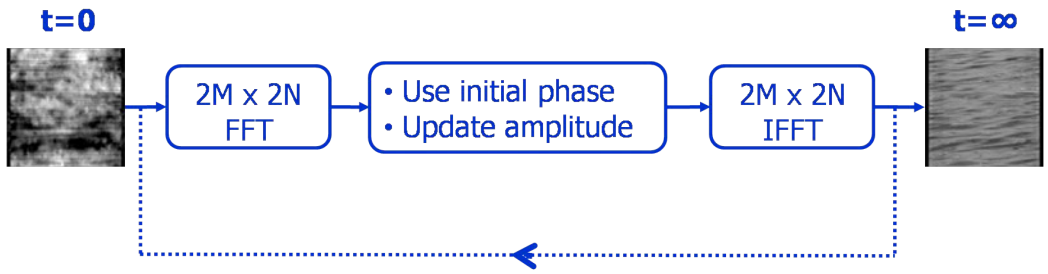
In what follows, we justify why the phase content of DT is a useful dual representation of its appearance and temporal variations, and why it leads to a compact spatiotemporal model. **(1)** In [20], Hayes proved that it is possible to reconstruct multi-dimensional signals from their phase content alone, provided that these signals do not have symmetric factors in their Z-transforms. In fact, if a hybrid image is constructed from the phase spectrum of a given image and the amplitude spectrum of any other, we use the iterative algorithm, described in [2], to reconstruct the original image from the hybrid image. This process is

called phase-only reconstruction and is shown schematically in Figure 2.1 (a). The initial phase response of the hybrid image is kept constant, while its amplitude response is updated iteratively through a sequence of FFT and inverse FFT (IFFT) operations of double the size of the hybrid image. At each iteration, only the first $M \times N$ (amplitude) pixels of the resulting image are retained. Figure 2.1(b) shows an example of this algorithm applied to ocean and fire images. In the rest of this paper, we assume that DT sequences possess this phase-only reconstruction property. This assumption is justified, since symmetric Z-transform factors seldom occur in practice.

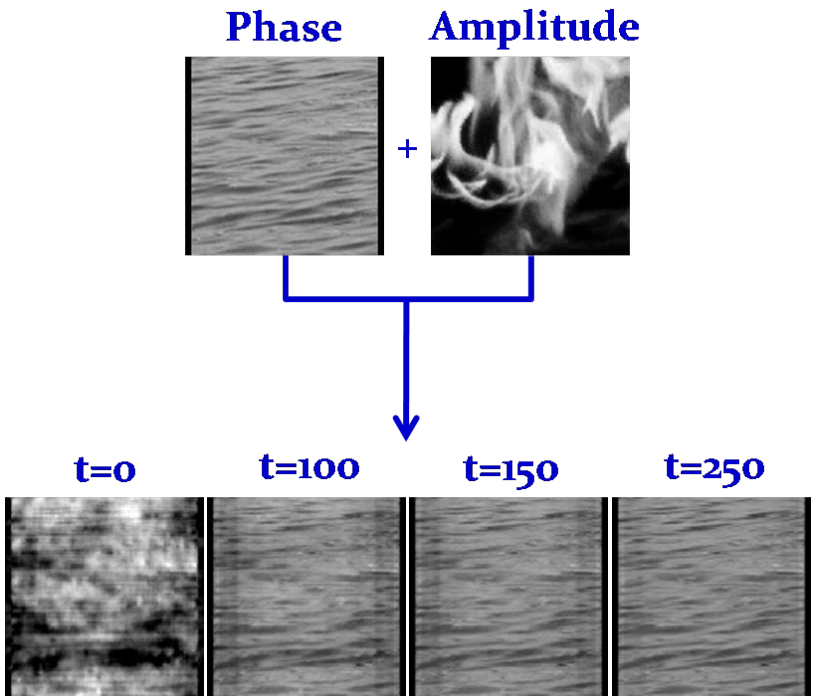
(2) Complex stochastic motion, which characterizes a DT, leads to complex stochastic variations in its phase content. We have empirically shown (refer to Figure 2.2), for a number of commonly encountered DT's, that the temporal variations of phase values do indeed capture most of the DT's dynamical characteristics and hence its global motion. This further validates, in addition to the phase-only-construction property, the value of phase for DT modeling. Figure 2.2 shows that many more principal components are required to represent 80% of the variation in the phase of a DT than to represent the same amount of variation in its amplitude. This implies that a DT's phase varies significantly more than its amplitude over time, and so the DT's dynamical properties are better captured in the Fourier phase space.

As a result of **(1)** and **(2)** above, we conclude that modeling a DT's global spatiotemporal features can be efficiently performed, solely in Fourier phase space.

Contributions: The contributions of the model we present in this thesis are two fold: **(1)** it exploits global spatial features via Fourier phase to form a computationally efficient spatiotemporal model of a DT's appearance and global dynamics, and **(2)** its training is insensitive to any static background of the DT, and hence does not require any segmentation or specialized cropping.



(a)



(b)

Figure 2.1: (a) shows a schematic of the iterative phase-only reconstruction algorithm described in [2]. This algorithm consists of a sequence of FFT and IFFT operations, where the phase response of the initial image ($t = 0$) is kept constant. In (b), this algorithm is applied to a hybrid image ($t = 0$), whose phase response comes from an ocean image and the amplitude response from a fire image. We show the intermediate results of this algorithm for iterations $t = 50, 100, 250$. Note that the reconstructed image significantly resembles the ocean image, from which the phase response was extracted.

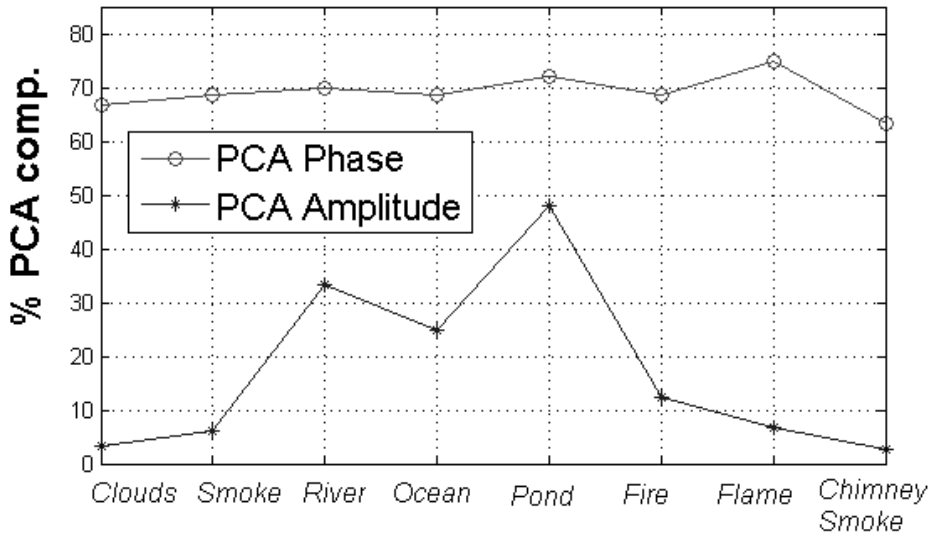


Figure 2.2: Percentage of PCA components required to capture 80% of the variation in DT phase and DT amplitude, for a set of DT video sequences. For each DT, it is clear that significantly more PCA components of the DT phase are needed than DT amplitude. This substantial discrepancy is due to the fact temporal variations of DT phase are much more significant than those of DT amplitude.

CHAPTER 3

PROPOSED MODEL

Our proposed model captures the phase portions of individual frames in a DT and, for efficiency, represents it in PCA space. We call this the Basic Phase PCA (BPP) model. To further increase the efficiency, we propose a model which captures the phase changes in DT over time, instead of the absolute phase values of each individual frame, as in BPP. We call this the Principal Difference Phase PCA (PDPP) model. PDPP represents the phase changes in terms of the principal angle of the difference between phase spectra of consecutive frames. We transform the BPP phase into PDPP format as follows. Each extracted phase spectrum ($\vec{\Phi}^{r+1}$) is vectorized and replaced by the sum of the previous phase spectrum ($\vec{\Phi}^{r}$) and the principal angle of their difference, as illustrated in Equation (3.1). For a DT sequence composed of F frames each of which is $M \times N$ pixels, we represent the $(r+1)^{\text{th}}$ phase spectrum as:

$$\Phi_i^{r+1} \longleftarrow \Phi_i^r + \Gamma(\Phi_i^{r+1} - \Phi_i^r) \;\; \forall i = 1, \cdots, MN \tag{3.1}$$

$$\Gamma(x) = x + 2\pi k \;\in\;]-\pi, \pi], \text{ for some } k \in \mathbb{Z}$$

In fact, this transformation expands the domain of the original BPP space by 2π in each dimension. Hence, the PDPP space can be spanned by fewer principal components, giving rise to a more compact spatiotemporal model. This can be better understood by considering the example of a simple rigid body translating in a video sequence with constant displace-

ment. In this case, the phase difference between every two consecutive frames is the same. Principal Difference Phase PCA (PDPP) dictates that each of the ordered DT phase spectra be replaced by the sum of the previous spectrum and the principal angle of the difference between the original spectrum and the previous one. If this is done, the feature vectors will be collinear in the high-dimensional space and a more compact PCA representation can be obtained. Figure 3 shows a 2D version of this situation, where A, B, and C represent the original feature vectors while A, $B^{'}$, and $C^{'}$ represent the transformed ones. Obviously, we can represent A, $B^{'}$, and $C^{'}$ in a lower dimensional space (i.e. the line connecting these collinear points) than the one required to span A, B, and C.

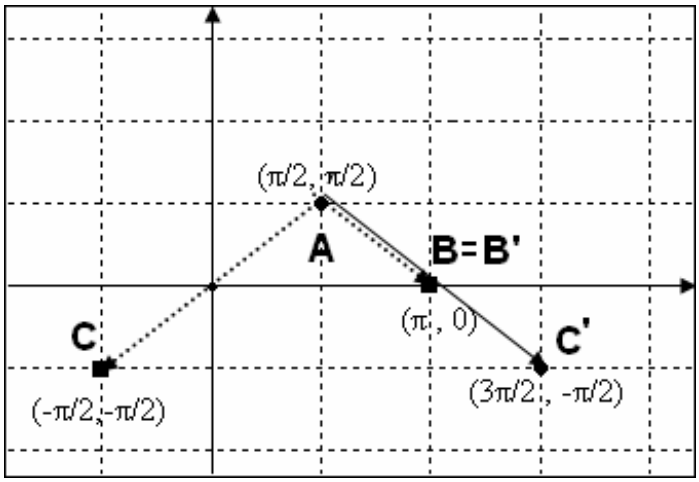
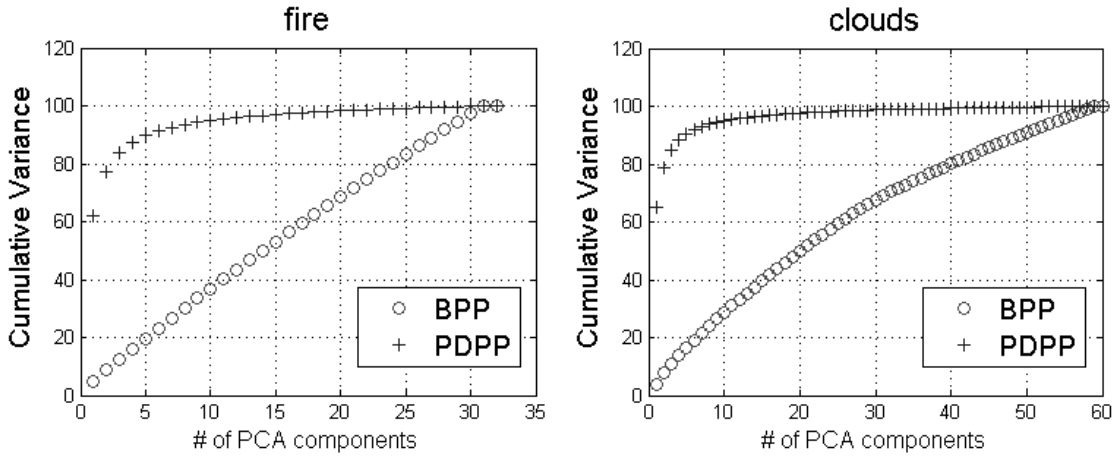


Figure 3.1: Schematic plot of BPP (A, B, and C) and PDPP (A, $B^{'}$, and $C^{'}$) features in 2D. Transforming BPP features into PDPP features expands the embedding range space by 2π in each dimension. Consequently, the number of PCA components needed to represent the PDPP space is, in general, less than the number needed to represent the BPP space. In this example, the BPP range space requires two PCA components, while the PDPP space requires only one.

In Figure 3.2, we show that, given a pair of equivalent PDPP and BPP models, the PDPP model is significantly more compact than the BPP one. In fact, Figure 3.2 provides empirical evidence that PDPP can capture significantly more variation in DT phase than BPP, for the same number of principal components. Consequently, a DT can be treated

as a sequence of features embedded in a low dimensional PDPP space. We use two different methods to represent the temporal variations in these features: either in a holistic manner, for all components, using a probabilistic framework, or modeling each component separately using a deterministic framework. In the rest of this chapter, we give a detailed description of both frameworks, and how the model can be applied to two major tasks involving DT video sequences: DT synthesis and DT recognition.



(a) (b)

Figure 3.2: (a) and (b) plot the PDPP and BPP cumulative variances represented by different numbers of PCA components for fire and clouds sequences. It is clear that the PDPP model represents much more cumulative variance than BPP for the same number of PCA components.

3.1 PDPP Algorithm

Each DT sequence is a set of F frames ($M \times N$ pixels in size). In order to mitigate spectral leakage, we preprocess the DT frames with a Hanning filter, whose spatial extent is set to be the image size. We extract the phase sub-spectra, whose energies exceed a predefined fraction of the total image energy. This fraction is increased to increase the compactness of the model. Only half of the phase spectrum is required due to Fourier conjugate symmetry. Next, we transform these sub-spectra into principal difference format.

Equation (3.2) illustrates how these spectra are embedded in a lower dimensional PCA space to form PDPP features ($\{\vec{x}^r\}_{r=1}^{F}$).

By reducing the number of principal components used in the representation to $L < F$, a more compact model is obtained. More importantly, a DT sequence containing a static background does not need to be cropped to show the dynamic texture alone. This follows from the fact that a static background will result in an additive term in the Fourier domain, which varies minimally with time. Since the PDPP features are formed from zero mean spectra and its basis spans the directions along which temporal variance is maximized (property of PCA), the effect of background on the model is highly reduced. Moreover, unlike LDS, no prior assumptions are made on the statistics of the DT sequence to be modeled.

$$\vec{\Phi}^r = A_{\text{PDPP}}\,(\vec{x}^r) + \vec{\Phi}_m \;\; \forall r = 1, \ldots, F \tag{3.2}$$

$$\vec{x}^r = A_{\text{PDPP}}^T(\vec{\Phi}^r - \vec{\Phi}_m) = [x_1^r \;\; x_2^r \;\; \cdots \;\; x_L^r]^T \tag{3.3}$$

$$\text{where } \vec{\Phi}_m = \frac{1}{F}\sum_{i=1}^{F} \vec{\Phi}_i \;\text{ and }\; L \leq F$$

We will now describe the two frameworks for learning the spatiotemporal manifold of a DT sequence in the PDPP feature space. The first framework (Section 3.2) captures the variations of all components in probabilistic terms, while the other (Section 3.3) captures the variations of each PDPP component separately and deterministically. Here, we note that forming images from phase is a nonlinear operation, since the phase appears in the complex exponent of the Fourier spectrum. So, for both frameworks, to achieve linearity we define the PDPP feature space to span the sinusoidal functions (i.e. cosine and sine) of the Fourier phase instead of the phase itself.

3.2 Non-Parametric Probabilistic PDPP Model

In this framework, a DT sequence (S) is represented as a set of F PDPP features, $\{\vec{x}_i\}_{i=1}^{F}$, corresponding to F frames. We model the posterior distribution of S non-parametrically, using Parzen windows with a suitably scaled unitary function $\Psi(\vec{x})$ as illustrated in Equations (3.4) and (3.5). We assume conditional independence between the features $\{\vec{y}_i\}_{i=1}^{L}$. We choose to use the RBF (radial basis function) kernel due to the simple functional form of its gradient and hessian.

$$p_S\left(\{\vec{y}_i\}_{i=1}^{L}|\{\vec{x}_m\}_{m=1}^{F}\right) = \prod_{i=1}^{L} p_S\left(\vec{y}_i|\{\vec{x}_m\}_{m=1}^{F}\right) \tag{3.4}$$

$$p_S\left((\vec{y}_i|\{\vec{x}_m\}_{m=1}^{F}\right) = \sum_{m=1}^{F} \frac{1}{V_F}\Psi_{RBF}\left(\frac{\vec{y}_i - \vec{x}_m}{h_F}\right) \tag{3.5}$$

$$\Psi_{RBF}\left(\vec{x}\right) = \frac{1}{2}e^{-\frac{\|\vec{x}\|^2}{2}};\;\; V_F = \left(h_F\right)^L = \frac{1}{\sqrt{F}}$$

We now describe how we use this probabilistic formulation for DT synthesis and recognition, using techniques from Bayesian machine learning.

3.2.1 MAP-Based DT Synthesis

We first consider using a given DT sequence to synthesize novel DT sequences, which resemble the original in appearance and global dynamics. In other words, given a set of PDPP features $\{\vec{x}_i\}_{i=1}^{F}$ that represent a DT, we want to find a new feature vector $\vec{x}_{MAP}$, which preserves the chosen spatiotemporal properties, namely of this DT. We formulate this synthesis problem as a multi-dimensional signal estimation problem according to the probabilistic framework described earlier. $\vec{x}_{MAP}$ is computed as the feature vector that maximizes the weighted posterior probability defined in Equation (3.6).

$$\vec{x}_{MAP} = \arg\max_{\vec{y}} \; p\left(\vec{y}|\{\vec{x}_i\}_{i=1}^{F}, \vec{w}\right) \tag{3.6}$$

$$p\left(\vec{y}|\{\vec{x}_i\}_{i=1}^{F}, \vec{w}\right) = \sum_{i=1}^{F} \frac{w_i}{V_F} \Psi_{RBF}\left(\frac{\vec{y} - \vec{x}_i}{h_F}\right)$$

$$\sum_{i=1}^{F} w_i = 1 \;\; ; \;\; w_i \geq 0 \;\; \forall i = 1, 2, \cdots, F$$

This optimization problem can be solved locally using Newton gradient descent to find $\vec{x}_{MAP}$, since it is an unconstrained, non-convex maximization problem. Using the RBF kernel simplifies the descent update stage. Different synthetic frames are produced when the frame MAP weights $\left(\{w_i\}_{i=1}^{F}\right)$ are varied. The impact of each original frame on the synthesis process is proportional to the magnitude of its corresponding weight. The larger the weight is, the more the synthetic frame resembles the corresponding original frame, in appearance, and dynamics relating it to the next frame. A DT sequence of arbitrary length can be synthesized by varying the MAP weights. This allows for extrapolation of appearance and dynamics of the original sequence without reproducing the original frames.

3.2.2 MAP-Based DT Recognition

We are given a set of C classes of DT, each of which contains DT sequences that have similar appearance and dynamical properties. A class c is represented by either a single PDPP model formed by concatenating all the DT instances in c or by a set of PDPP models (each for a different DT in c) that will be processed independently. Experimentation shows us that both methods result in similar recognition rates. For the sake of simplicity, let us assume that each trained model (c) is represented by a single DT sequence, $\{\vec{y}_i^c\}_{i=1}^{F}$.

Each given test sequence (S_T) of T frames is represented by a sequence of T features $\{\vec{x}_i^c\}_{i=1}^{T}$ for each class c. In other words, $\{\vec{x}_i^c\}_{i=1}^{T}$ are the projections of S_T onto the PDPP

space that spans $\{\vec{x}_i^c\}_{i=1}^T$ and $\{\vec{y}_i^c\}_{i=1}^F$. Therefore, the task of recognizing S_T becomes the task of finding the class c^*, which maximizes the posterior probability of $\{\vec{x}_i^c\}_{i=1}^T$ over all classes. This is formulated as follows:

$$c^* = \arg\max_c \; p\left(\{\vec{x}_i^c\}_{i=1}^T | \{\vec{y}_k^c\}_{k=1}^F\right) \tag{3.7}$$

$$p\left(\{\vec{x}_i^c\}_{i=1}^T | \{\vec{y}_k^c\}_{k=1}^F\right) = \prod_{i=1}^T p\left(\vec{x}_i^c | \{\vec{y}_k^c\}_{k=1}^F\right)$$

$$p\left(\vec{x}_i^c | \{\vec{y}_k^c\}_{k=1}^F\right) = \sum_{k=1}^F \frac{1}{V_F} \Psi_{RBF}\left(\frac{\vec{x}_i^c - \vec{y}_k^c}{h_F}\right)$$

3.3 Piecewise Smooth PDPP Model

Local correlations between neighboring frequencies in the Fourier phase domain are considerably smaller than those between neighboring pixels in the spatial domain. Making use of this property, we assume that the components of a PDPP feature can be modeled as being independent. In this regard, each component is represented by a temporally varying trajectory, which is inherently oscillatory. Consequently, we choose to model the trajectory of the m^{th} PDPP component as a piecewise smooth function, $f\left(t|\vec{\theta}_m\right)$, (e.g. spline). Using the independence assumption among the components, the PDPP model of a DT can then be viewed as a sequence of samples from L independent models given in Equation (3.8). Although the component-wise independence assumption neglects underlying component correlations, it allows for a more compact and computationally efficient model, as compared to the probabilistic model described earlier.

$$x_m(t) \triangleq f\left(t|\vec{\theta}_m\right) \quad \forall t \geq 0;\ \forall m = 1, \ldots, L \tag{3.8}$$
$$\vec{\theta}_m = \arg\min_{\vec{\theta}} \sum_{i=1}^{F} [f\left(i|\vec{\theta}\right) - x_m^i]^2$$

For DT synthesis, we choose $f\left(t|\vec{\theta}_m\right)$ to be a cubic spline (i.e. piecewise cubic polynomials), which is sampled at equal intervals. Continuity and smoothness constraints (e.g. consecutive cubic pieces must have equal 1^{st} and 2^{nd} order derivatives where they meet) must be incorporated in estimating $\vec{\theta}_m$. Figure 3.3 illustrates the trajectory of a PDPP feature component modeled as a cubic spline. Learning this trajectory facilitates the interpolation and extrapolation of novel DT frames.

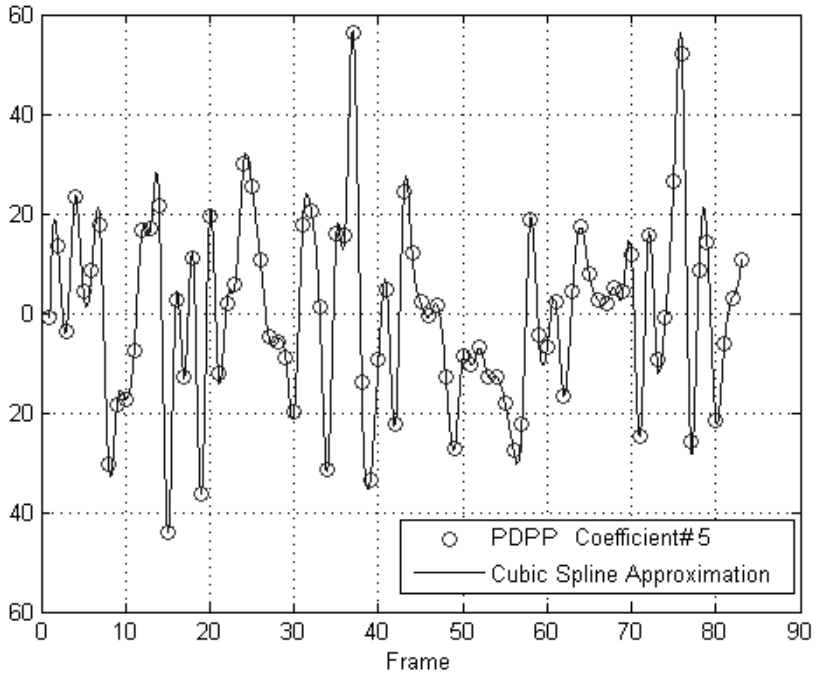


Figure 3.3: Temporal variation of a PDPP feature component modeled by a cubic spline.

CHAPTER 4

DT VIDEO COMPRESSION

In this chapter, we employ the dimensionality reduction property of PCA to increase the compactness of the PDPP model, thus, rendering a compression scheme for DT video sequences.

For DT video compression, we perform PCA on the DT feature vectors, which are the vectorized half spectra of DT phase for the frames in the DT sequence. For a DT frame of size $M \times N$, the length of the feature vector is $K = \frac{MN}{2}$, so that the size of A_{PDPP} is $K \times L$, where L is the number of principal components that have been selected to represent the data. For complete representation, $L = F$, where F is the number of frames in the DT sequence.

Due to the symmetry of DT phase, considerable compression is possible for each individual frame. We note here that additional compression can be achieved by neglecting frequencies with low energy, mainly in the high frequency bands and by using the dimension reduction option inherent to PCA. Below, we present the compression rates that can be achieved by PDPP and LDS in terms of the number of principal components used for DT representation.

4.1 PDPP Compression

In this section, we will present the overall compression rate that can be achieved by reducing the dimensionality of the PDPP space. We can compute the expected overall compression rate (R_{PDPP}) for an arbitrary DT sequence as shown in Equation (4.1). In fact, since

$L \leq F$ and $\frac{MN}{2} \gg F$, then the removal of a PCA component will lead to significant data compression. Since $L \ll MN$, we have

$$R_{\text{PDPP}} = 1 - \frac{\text{size}(A_{\text{PDPP}}) + \text{size}(\vec{\Phi}_m) + \text{\# of PCA coefficients}}{MNF}$$

$$R_{\text{PDPP}} = 1 - \left[\frac{L+1}{2F} + \frac{L}{MN}\right] \approx 1 - \frac{L+1}{2F} \tag{4.1}$$

The main factor dictating the extent of the data compression is $\frac{L}{F}$, the fraction of the PCA components used in the representation. Also, note that even with a complete representation ($L = F$), the compression rate is about 50%. This is due to the fact that only half the phase spectrum is used to represent the DT, so the amplitude spectrum must be initially determined by the iterative process mentioned in the context of phase-only reconstruction (refer to Figure 2.1(a)).

4.2 LDS Compression

Using LDS, we require two matrices (**A** and **C**) and the initial state $\vec{x}_0$ in order to reconstruct the DT. The dimension of **A** is $L^{'} \times L^{'}$ and that of **C** is $2K \times L^{'}$, where $L^{'}$ represents the order of the LDS and $K = \frac{MN}{2}$ as defined before. So, the overall compression rate (R_{LDS}) is estimated as in Equation(4.2). Note that under the same compression rate, the LDS method requires approximately half the number of principal components needed by PDPP.

$$R_{\text{LDS}} = 1 - \frac{\text{size}(\mathbf{A}) + \text{size}(\mathbf{C}) + \text{size}(\vec{x}_0)}{MNF}$$

$$R_{\text{LDS}} = 1 - \frac{\left(L^{'}\right)^{2} + 2K \times L^{'} + L^{'}}{MNF} \approx 1 - \frac{L^{'}}{F} \tag{4.2}$$

From Equations (4.1) and (4.1), we note that our model provides at least twice as much compression as an equivalent LDS with the same model order. In the next chapter, we will see that this discrepancy in compression performance becomes more significant when both methods are applied to DT sequences. This is primarily attributed to the difference in encoding schemes exploited by each method.

CHAPTER 5

EXPERIMENTAL RESULTS

In this chapter, we will illustrate the performance of the PDPP model when applied to three applications: **(I)** DT synthesis, **(II)** DT recognition, and **(III)** video compression of DT sequences. The probabilistic and component-wise models are used in synthesis, while only the former is used in recognition. We will compare our results with those of LDS for all three applications, in addition to a comparison with the MPEG-2 encoding scheme.

5.1 APPLICATION (I): DT Synthesis

Numerous techniques have been proposed for DT synthesis. Some model the physical process underlying the DT (e.g. formation of ocean waves) [7]. Despite their high visual quality, the specificity of these models prevents them from being generalized to other DT's. As an alternative to physical models, purely image-based approaches have also been developed. In this category, we distinguish between two main groups: the first does not formulate a model of the DT but instead it reuses real frames from various locations in the sequence to extend the original sequence, while maintaining smooth frame-to-frame transition [21]. The other group of methods synthesizes frames based on a learned model of the DT [4,9,10,11]. Among the few such model-based techniques that have been proposed, LDS has received the most attention in recent work. Despite its succinct representation, its main assumptions (e.g. second order stationarity, linearity in the spatial domain, and suboptimal temporal modeling) limit the visual quality of synthesized DT sequences, es-

pecially non-stationary ones (e.g. fire). For such DT sequences, the visual quality of the synthetics frames deteriorates over time.

To evaluate PDPP based synthesis, we compiled a database of DT sequences from various online sources including the recent DynTex database [22]. MATLAB implementations for both of our PDPP methods were developed. For the MAP-based method, only two consecutive MAP weights were set to nonzero values. For the cubic spline method, we used equal length sampling intervals. Figure 5.1 shows some images randomly sampled from synthetic sequences produced by both PDPP models.

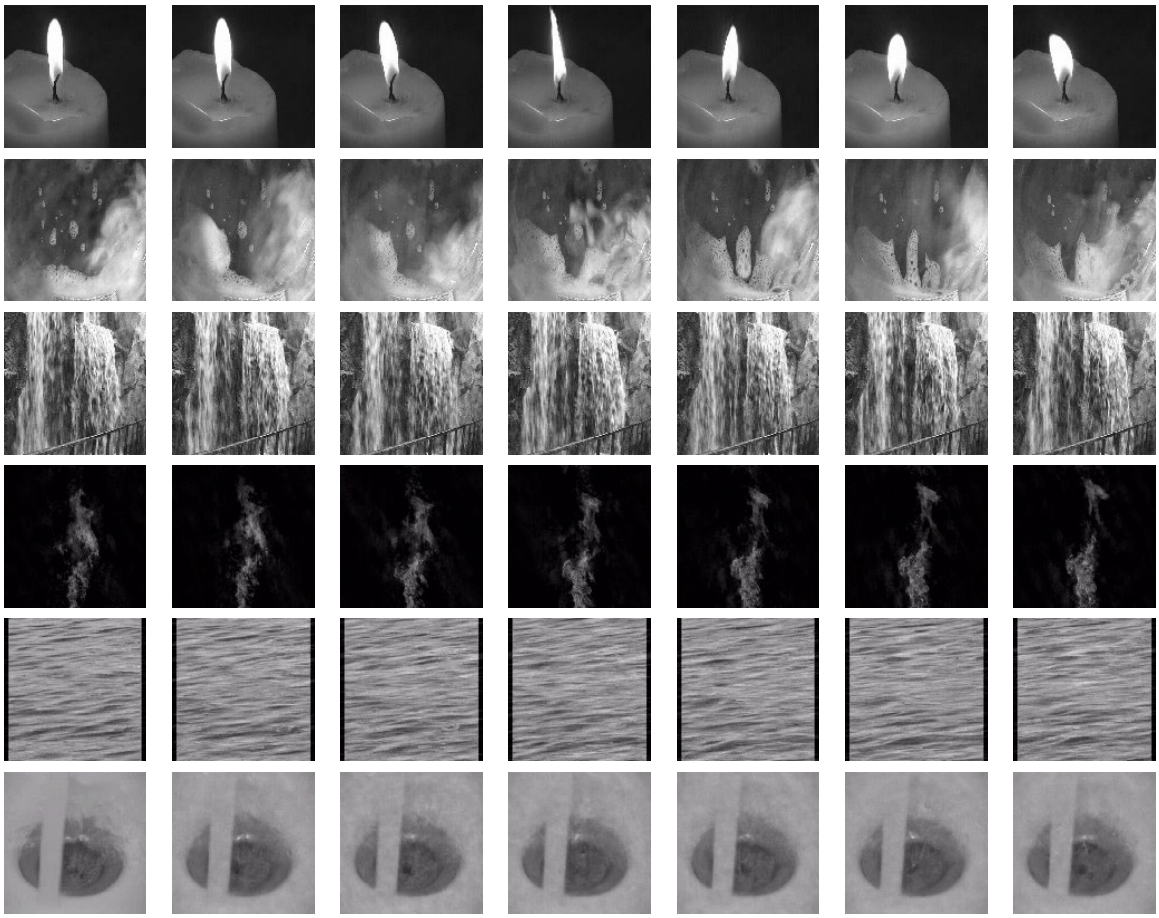


Figure 5.1: Sample synthetic images produced by the Cubic Spline (top three rows) and MAP (bottom three rows) synthesis methods. Videos of these sequences can be viewed at [3].

EVALUATION

We compare the quality of our MAP synthesis with that of LDS for the same DT sequence. The dimensionality of the LDS model was set to be the same as that of our PDPP model. In Figure 5.2, we show an example of DT synthesis for a flame sequence. The quality of the synthetic frames appears to be more "natural" for PDPP than for LDS, since PDPP better preserves DT boundaries compared to LDS, which tends to blur them. Note that PDPP maintains smooth frame-to-frame transition, while undergoing global dynamics that resemble the original sequence.

Following observations can be made about the synthesis quality of PDPP and LDS. **(1)** Frames synthesized by LDS tend to be blurred (e.g. (c)) caused by LDS' underlying linear, spatial model. PDPP does not suffer from this problem; however, some synthetic frames contain spatially periodic noise due to residual spectral leakage. **(2)** For some DT sequences, LDS produces synthetic frames whose visual quality degrades with time. For PDPP, the temporal quality degradation is considerably less. **(3)** As the order of the PDPP or LDS model decreases, the visual quality (i.e. appearance and global dynamics) of the synthesized frames degrades for both; however, the LDS frames tend to display significantly less temporal variation than PDPP. This follows from the fact that an $n < F$ dimensional PCA basis captures the maximum variance, among all n dimensional bases, of the same set of feature points.

5.2 APPLICATION (II): DT Recognition

DT recognition involves the use of both image appearance and temporal changes in appearance. For an overview of recent techniques developed for DT recognition, we refer to [23]. In [12], Doretto et al. use the LDS model parameters of each DT to recognize them. Fujita et al. use impulse responses of state variables as alternative features for recognition using

(a) (b) (c)

Figure 5.2: Images (a), (b), and (c) are respectively a frame from the original sequence, a MAP synthetic frame, and a synthetic LDS frame.

the LDS model [24]. In [25], Peteri et al. propose a DT recognition algorithm based on six translation invariant features (i.e. normal optical flow and texture regularity to describe DT dynamics and appearance respectively). Recent work by Zhao et al. proposes local binary patterns (LBP) and volume local binary patterns (VLBP) as the underlying features to be used in recognizing DT sequences [26,27]. The latter two methods are based on local descriptors, which do not incorporate the global dynamics that characterize a dynamic texture. All the above algorithms have been evaluated on subsequences or sub-blocks of the initial sequences on which the model was trained. It is unclear how they will perform on test DT sequences not already used in training. Since LDS is the model that has been used the most and seems to have the best performance among image-based models, in what follows, we compare the performance of our MAP-based recognition method to that of LDS and evaluate its generalization performance.

EVALUATION

We constructed a database of color DT sequences from the DynTex database and online sources. These sequences portray natural scenes including different bodies of water (rivers, oceans, waterfalls, etc. as in Figure 5.1), fire, foliage, clouds, and smoke. These DT's possess a variety of appearance and dynamics characteristics, which is a significantly richer

and more challenging environment for testing our recognition algorithm as compared to the MIT temporal texture database [28]. In our database, similar looking textures (e.g. fire) may have different dynamics, while some different looking textures (e.g. smoke, water, and fire) may have similar dynamics. These DT's have different sizes and number of frames $(40 - 250)$. To expedite FFT, images in each sequence are resized to 128×128 pixels and converted to gray scale format. However, no cropping is performed. We formed two groups of DT subsequences for this purpose: the first group contains a subsequence of the frames of these formatted sequences for training the PDPP model, while the second group contains subsequences formed by randomly choosing consecutive frames that were not used for training and do not overlap. The latter subsequences form our test set.

In our experiments, we vary either: F_{train} (number of frames used to train the PDPP model) or F_{test} (number of frames in the test subsequence). In each case, PDPP performance is compared to that of LDS, based on the implementation of [12]. The overall recognition rates for LDS vs. PDPP are included in Table 5.1.

F_{train}	F_{test}	**LDS** (%)	**PDPP** (%)
20	20	43.8	78.8
30	30	62.4	89.5
$\frac{F}{2}$	10	35.6	90.8
$\frac{F}{2}$	20	**46.3**	**95.2**

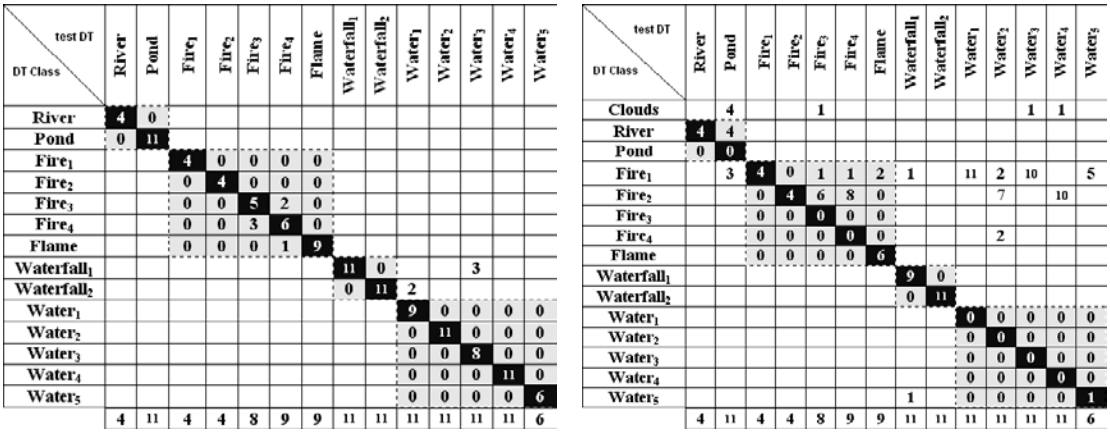
Table 5.1: Recognition rates of LDS vs. PDPP for different training and test settings. F_{train} denotes the number of frames used for training the PDPP model. F_{test} denotes the number of frames in each test sequence.

Because of space restriction, we only show the detailed results of one of these experiments (i.e. last row of Table 5.1), where 147 test subsequences were formed from $C = 17$ different DT classes from the database. F_{train} is equal to the first half of the C original sequences and $F_{\text{test}} = 20$ frames. Figure 5.3 shows the confusion matrices for both PDPP and LDS. The columns represent the labeled test subsequences, while the rows represent the corresponding recognition results. The recognition is deemed correct if the recognized

class is among the types shaded in gray. For example, of the 9 "flame" test subsequences, LDS recognizes 6 as "flame" and 2 as "fire$_1$", while PDPP recognizes them all as "flame". We obtained overall recognition rates of 95.2% and 46.3% for PDPP vs. LDS. Interestingly, the LDS method finds difficulty in distinguishing between the "fire" and "water" sequences. This shows the greater discriminating power of PDPP. Note that if we were to accept only diagonal entries as correct recognition, then the performance disparity between PDPP and LDS would be greater.

Based on the results of these experiments, we draw the following conclusions. **(1)** The recognition performance of LDS improves as F_{test} becomes comparable to F_{train}. In the previous detailed experiment, $F_{\text{test}} \ll F_{\text{train}}$, as compared to [12] where $F_{\text{train}} = F_{\text{test}} = 75$. This follows from the nature of the distance metric used (Martin distance [29]), which requires that the orders of the training and test LDS models be the same, so the test LDS model has to be expanded to the same size as the training model, as described in [30]. On the other hand, PDPP naturally accommodates any sized test sequence. **(2)** The recognition performance of both methods is directly proportional to F_{train}. However, the performance change is more significant for LDS, which means that LDS, in general, requires a larger training set than PDPP for comparable recognition rates. **(3)** The presence of a static background in the training and/or test sequences decreases the recognition rate for LDS considerably, since the LDS model does not distinguish between DT and background properties. This follows from its direct modeling of pixel intensities in the spatial domain. **(4)** A major drawback of the LDS model is its memory usage and computational complexity. In fact, all our experiments required $F_{\text{test}} \leq 30$ frames in order to run on a Pentium IV (2GB RAM) PC and keep the running time of the LDS algorithm less than 3 minutes per test sequence; otherwise, the recognition process ran out of memory. PDPP does not face this problem. For $F_{\text{test}} = 30$ frames, LDS runs in approximately 3 minutes, while our algorithm runs in about 30 seconds for the same test sequence.

Unlike the DT recognition methods we referred to earlier, we tested the generalization performance of PDPP with $C = 4$ categories (i.e. smoke, fire, water, and grass). The experiment was set up as follows: one sequence was used to learn the "smoke" class, two for "fire", five for "water", and one for "grass". A total of 141 test subsequences were formed from 7 new DT sequences, which did not participate in the training stage. Table 5.2 summarizes the recognition results. We conclude that the PDPP model achieves better generalization performance than LDS.



(a) (b)

Figure 5.3: (a) and (b) are portions of the confusion matrices for recognition of column DT as row DT by PDPP and LDS respectively. Blank entries correspond to no decisions.

DT	**LDS** (%)	**PDPP** (%)
Smoke	100	100
Fire	87.5	100
Water	77	81
Grass	44.4	100
Weighted Average	**61**	**87**

Table 5.2: Recognition rates of LDS vs. PDPP on distinct DT categories

5.3 APPLICATION (III): DT Compression

In this section, we present experimental results that validate the significance of our proposed method for DT compression and compare its performance to that of LDS and MPEG-2 encoding.

EVALUATION

Figure 5.4 (a)-(c) compare the performance of LDS, BPP, and PDPP over a range of compression rates, that are proportional to the number of principal components used. We note here that the compression rate is computed from the number of principal components required by LDS. It is evident from these plots that BPP, in general, outperforms the LDS compression scheme, mainly due to the fact that only half the phase spectrum is modeled. Furthermore, PDPP renders a significant improvement over BPP even at very low compression rates. In Figure 5.4(d), we show the temporal performance of each compression scheme at a compression rate of around 60%. We notice that both BPP and PDPP tend to oscillate about a steady PSNR value, while LDS performance decreases with time. This is due to the fact that LDS produces a DT frame as a linear combination of the $L^{'}$ chosen principal components, which are computed from $L^{'}$ frames of the original DT and not all F of them.

Next, we compare PDPP compression to that of the MPEG-2 standard, as portrayed in Figure 5.4 (e)-(h). Here, we define the compression rate for each case as the ratio of the size of the MPEG-2 video to that of the original, uncompressed video. This was set to approximately 60% in each case. From the above plots, we see that our method outperforms MPEG-2 in all four DT's. This improvement is primarily due to the more compact representation of the temporal characteristics of the DT inherent to PDPP. Since MPEG-2 requires computation of motion fields and these estimates based on optical flow algorithms

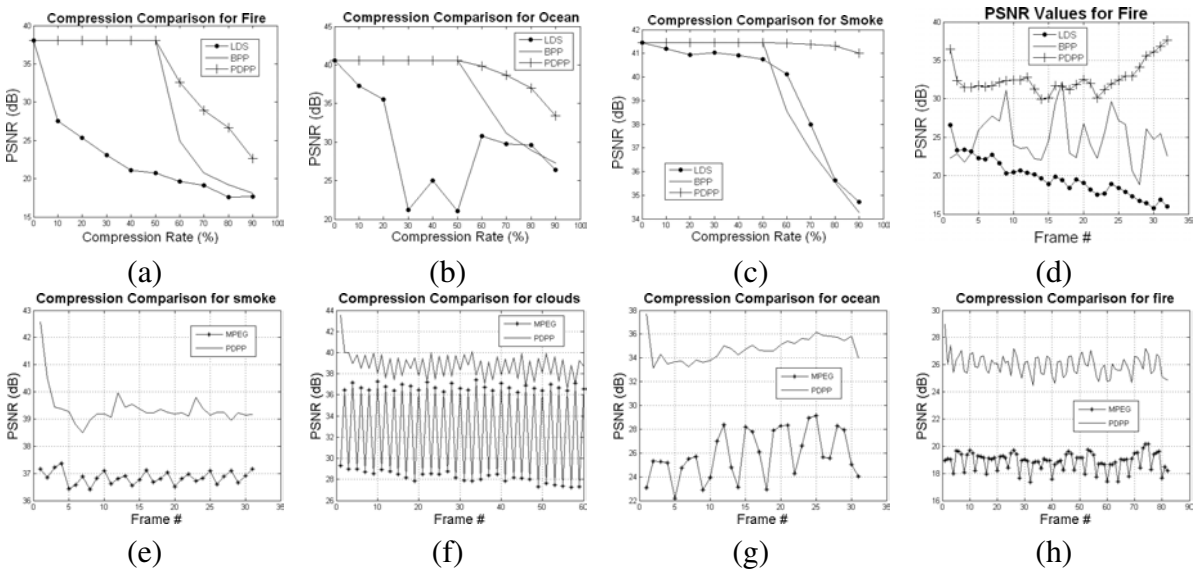


Figure 5.4: (a)-(c) compare the mean compression performance of LDS, BPP, and PDPP for fire, ocean, and smoke DT video sequences. In (d), we show the temporal variations of LDS, BPP, and PDPP compression schemes for the fire sequence. In (e)-(h), we compare the compression performance of our PDPP algorithm to that of MPEG-2 encoding for a compression rate of approximately 60%.

tend to degrade with the complexity of the motion and the moving objects, MPEG-2 does not perform as well for the stochastic motion of non-rigid particle objects prevalent in DT's.

CHAPTER 6

CONCLUSIONS

In this thesis, we presented a novel spatiotemporal model (PDPP) for dynamic textures, which is based on the variation of Fourier phase content. PDPP compactly and efficiently represents both the appearance and dynamics of a DT. We have validated the significance of our method by applying it to three applications: DT synthesis, DT recognition, and video compression of DT sequences. In fact, we show empirical evidence that our PDPP model outperforms the LDS model in these applications. For the compression application, we show that this model can render a compression scheme capable of outperforming MPEG-2 when applied to DT video sequences. Based on the theoretical and empirical premises established in this thesis, the PDPP model comprises a general framework for DT analysis, thus, rendering it flexible enough to accommodate for various high-level applications.

CHAPTER 7

FUTURE WORK

Despite its fundamental advantages over the LDS model, the current PDPP model suffers from shortcomings that can be overcome via future research endeavors, as described in what follows.

- In MAP-based DT synthesis, PDPP does not explicitly enforce temporal coherence in the MAP reconstruction algorithm. Coherence can be achieved by formulating the synthesis problem into a graph cut (i.e. MRF) problem. In this Markovian framework, synthetic frames are no longer optimized independently, yet they are inherently correlated.

- PDPP currently processes a single DT phenomenon in each sequence. We plan to develop a PDPP-based method for mixture modeling to handle more than one DT in a sequence.

- PDPP assumes that a single linear subspace can sufficiently represent the DT manifold. We will experiment with multiple linear models, in order to improve recognition and synthesis performance, while maintaining its generalization capabilities.

- We plan to extend the applications of our PDPP framework to DT segmentation, where a DT can be successfully extracted from its embedding background.

REFERENCES

[1] http://vision.ai.uiuc.edu/~bghanem2/Shared/DTdatabase/.

[2] A. V. Oppenheim and J. S. Lim, "The importance of phase in signals," *Proceedings of the IEEE*, vol. 69, pp. 529–541, 1981.

[3] http://vision.ai.uiuc.edu/~bghanem2/Shared/DTSynthesis/.

[4] R. Nelson and R. Polana, "Qualitative recognition of motion using temporal texture," in *Proc. of the International Conference on Pattern Recognition*, 1992, pp. 56–78.

[5] P. Bouthemy and R. Fablet., "Motion characterization from temporal cooccurrences of local motion-based measures for video indexing," in *Proc. of the International Conference on Pattern Recognition*, vol. 1, 1998, pp. 905–908.

[6] D. Comanicui, V. Ramesh, and P. Meer, "Kernel-based object tracking," in *IEEE Trans. in Pattern Analysis and Machine Intelligence*, vol. 25, 2003, pp. 564–575.

[7] A. Fournier and W. Reeves, "A simple model of ocean waves," in *Proc. of ACM SIGGRAPH*, 1986, pp. 75–84.

[8] B. H. McCormick and S. N. Jayaramamurthy, "Time series model for texture synthesis," *International Journal of Computer and Information Science*, vol. 3, pp. 329–343, 1974.

[9] M. Szummer and R. W. Picard, "Temporal texture modeling," in *Proc. of the International Conference on Image Processing*, vol. 3, 1996.

[10] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, "Texture mixing and texture movie synthesis using statistical learning," *IEEE Trans. on Visualization and Computer Graphics*, pp. 120–135, 2001.

[11] S. Soatto, G. Doretto, and Y. N. Wu, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, pp. 91–109, 2003.

[12] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto, "Dynamic texture recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2001, pp. 58–63.

[13] G. Doretto, D. Cremers, P. Favaro, and S. Soatto, "Dynamic texture segmentation," in *Proc. of International Conference on Computer Vision*, vol. 2, 2003, pp. 1236–1242.

[14] A. B. Chan and N. Vasconcelos, "Mixture of dynamic textures," in *Proc. of the International Conference on Computer Vision*, vol. 1, 2005, pp. 641–647.

[15] B. Abraham, O. I. Camps, and M. Sznaier, "Dynamic texture with fourier descriptors," in *Proc. of International Workshop on Texture Analysis and Synthesis*, 2005, pp. 53–58.

[16] R. Li, T. Tian, and S. Sclaroff, "Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series," in *Proc. of the International Conference on Computer Vision*, 2007.

[17] J. Huang, X. Huang, and D. Metaxas, "Optimization and learning for registration of moving dynamic textures," in *Proc. of the International Conference on Computer Vision*, 2007.

[18] M. B. Savvides, V. Kumar, and P. Khosla, "Eigenphases and eigenfaces," in *Proc. of the International Conference on Pattern Recognition*, vol. 3, 2004, pp. 810–813.

[19] A. Briassouli and N. Ahuja, "Spatial and fourier error minimization for motion estimation and segmentation," in *Proc. of the International Conference on Pattern Recognition*, vol. 1, 2006, pp. 94–97.

[20] M. Hayes, "The reconstruction of a multidimensional sequence from the phase or magnitude of its fourier transform," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 30, no. 2, 1982.

[21] A. Schodl, R. Szeliski, D. Salesin, and I. Essa, "Video textures," in *Proc. of ACM SIGGRAPH Conference*, vol. 25, 2000, pp. 489–498.

[22] R. Peteri, M. Huiskes, and S. Fazekas, DynTex: www.cwi.nl/projects/dyntex/ at the Centre for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands.

[23] D. Chetverikov and R. Peteri, "A brief survey of dynamic texture description and recognition," in *Proc. of the International Conference on Computer Recognition Systems*, 2005.

[24] K. Fujita and S. K. Nayar, "Recognition of dynamic textures using impulse responses of state variables," *Proc. Of Third International Workshop on Texture Analysis and Synthesis*, pp. 31–36, 2003.

[25] R. Peteri and D. Chetverikov, "Dynamic texture recognition using normal flow and texture regularity," in *Proc. of the Iberian Conference on Pattern Recognition and Image Analysis*, 2005.

[26] G. Zhao and M. Pietikainen, "Dynamic texture recognition using volume local binary patterns," in *Proc. of the European Conference on Computer Vision (Workshop on Dynamical Vision)*, 2006, pp. 12–23.

[27] ——, "Local binary pattern descriptors for dynamic texture recognition," in *Proc. of the International Conference on Pattern Recognition*, vol. 2, 2006, pp. 211–214.

[28] ftp://whitechapel.media.mit.edu/pub/szummer/temporal-texture/.

[29] R. J. Martin, "A metric for arma processes," *IEEE Trans. on Signal Processing*, vol. 48, pp. 1164–1170, 2000.

[30] K. DeCock and B. De Moor, "Subspace angles between ARMA models," in *Systems and Control Letters*, vol. 46, 2002, pp. 265–270.